AD-A048 124   CLEMSON UNIV S C DEPT OF MATHEMATICAL SCIENCES   F/G 9/2

CONSTRAINTS IN THE DESIGN AND IMPLEMENTATION OF INTERACTIVE STA--ETC(U)

MAY 77 R F LING   N00014-75-C-0451

UNCLASSIFIED   N84   NL

1 OF 1

AD

A048124

END

DATE

FILMED

1-78

DDC

(12)

(See backpage for 1473)

DEPARTMENT OF MATHEMATICAL SCIENCES

CLEMSON UNIVERSITY
Clemson, South Carolina

CONSTRAINTS IN THE DESIGN AND IMPLEMENTATION OF INTERACTIVE STATISTICAL SYSTEMS FOR MINICOMPUTERS

BY

ROBERT F. LING

N 84

TECHNICAL REPORT # 251

MAY 1977

To appear in Proceedings of Computer Science and Statistics: 10th Annual Symposium on the Interface, National Bureau of Standards, Gaithersburg, Md., 1977.

Research supported in part by ONR Contract N00014-75-C-0451

1. INTRODUCTION

During the past decade, the minicomputer industry has experienced an explosive period of growth, in terms of technological advances and market volume. According to recent Datapro Research Corporation Reports, estimates of worldwide minicomputer market volumes are:

1972 [1] $300 - $450 million

1975 [2] $800 million - $1.4 billion

1977 [2] $1.8 billion.

These figures are rather striking by themselves even if we do not take into account the rapid decrease in the cost of central processors. Kenney [10] wrote, "In 1966, for example, the processor cost approximately $30,000, but six years later, 1972, its price was only 20 percent of that cost, about $6,500." Monrad-Krohn [12] (1977) estimated, "The central processing element of a computer has decreased to the cost of about $20."-- of course, he was referring to the lower spectrum of present generation of micro computers.

During this period of explosive growth, technological advances in the hardware components have far exceeded the development of software. The following quotes are fairly typical of current opinions about minicomputer software:

> "The present state of software development is far from being acceptable ... Development of the software takes longer than anticipated and almost always the costs are more than expected. At times the finished product does not perform as expected, and there have been times when it didn't perform at all." [10, p. 76]

"Software, which had long received only cursory attention form the predominantly hardware-oriented minicomputer makers, is rapidly becoming the principal distinghishing factor between competitive product lines." [2, p 70c-010-20d]

Given the state of general software development of minicomputers, it should be no surprise that existing statistical software for minicomputers is fragmented, localized, and often primitive. Some manufacturers (such as Hewlett-Packard) serve as the distributor of user-contributed software, including statistical programs and systems. In such cases, the lack of quality control standards for contributed programs resulted in many library programs that are low in quality, by any reasonable standards of evaluation. Portable statistical systems for minicomputers, interactive or not, are almost nonexistent. MiniBMD [5] is perhaps the first serious attempt at the creation of a portable, high quality, general purpose statistical system specifically designed for minicomputers.

For the aforementioned reasons, instead of doing a survey of existing, non-portable, statistical software, I shall consider some characteristics of portable statistical software for minicomputers in the immediate future by focussing on constraints imposed by such computers on the design and implementation of interactive statistical systems. In my opinion, interactive systems are of paramount importance in the effective use of statistics on minicomputers, and the effective design of such systems must pay close attention to the constraints.

2. WHAT IS A MINICOMPUTER?

One agreement within the minicomputer industry is that there is disagreement as to what constitutes a minicomputer. For the purpose of the present discussion, I shall use the pseudo-definition "minicomputers are machines whose mainframes sell for less than $50,000 (or some other arbitrary figure)" in the spirit minicomputers are defined in [2]. A typical system configuration costs two to four times the cost of the mainframe. There are no clear cutoff values that separate minis from micros and midis (see e.g. [12, 15]). For example, Interdata 8/32 is classified as a mini in [2] and a midi in [15]. Given the trend of increasing computer power and decreasing cost, the next generation of minis will likely be comparable to some of today's maxis in capacity and performance.

The most important distinguishing characteristic of a mini is its word length. A "typical" mini currently on the market has a 16-bit word length, although minis with word lengths of as many as 32 bits or as low as 8 bits are not rare. For a minicomputer which is capable of supporting a moderately versatile interactive statistical system, we may consider the following to be some of its "typical" characteristics:

Software support: a time sharing operating system. BASIC and/or FORTRAN compilers.

Main Format: 16-bit word length (and up).

Main storage: magnetic core having a maximum storage capacity of 32768 words (and up).

I/O control: DMA channel and multilevels of external interrupt.

Peripheral: disk pack or cartridge drives, tape deives and other standard I/O devices.

3. CHOICE OF COMPUTER AND INTERACTIVE SYSTEM DESIGN -- WHICH COMES FIRST OR SHOULD IT MATTER?

From the system designer's point of view, two general optimization approaches are possible:

(A) Consider an ideal design of an interactive system and then choose a computer whose characteristics are most suitable for the implementation of that design.

(B) Given a computer and its associated software, design an interactive system which attempts to make optimal use of the available features and resources.

In practive, approach (A) is generally not available to the statistical system designer; and judging from the characteristics of existing interactive statistical software for large and small computers, approach (B) appears to be the norm. As a result, most of them (e.g., IDA [11], isp [4], MIDAS [6, 7], SAS [13], SIPS [9], and SPEAKEASY [14]) achieve certain desirable features or local optimality at the expense of severely limited portability.

If we use the criteria for evaluating statistical software in [8, 16] as guidelines for designing an interactive system, then neither approach (A) nor approach (B) would be appropriate. Instead, the system designer should first consider the constraints imposed by the requirement of portability to choose the software language used to code the interactive system (e.g., at the present time, neither APL nor PL/I would be an appropriate choice because most minicomputers do not have an interpreter or compiler for these languages, although purely from a programming language point of view, they are in many respects better than their counterparts BASIC and FORTRAN which are widely supported.)

Our experience with existing interactive systems should have taught us a lesson about the importance of portability. Far too often, system designers (myself included) exhibit systems with many desirable features but unfortunately have to inform those who are interested in using the system that it cannot run under machine ABC or operating system XYZ without substantial conversion efforts. In order to consider a truly portable system, we are not only constrained to use BASIC or FORTRAN, but we must sacrifice certain features of a system if their implementation would require non-standard features of those languages. Similarly, other constraints imposed by minicomputers should be carefully considered before a system is designed or implemented.

4. CONSTRAINTS IMPOSED BY MINICOMPUTERS

The major categories of evaluation criteria and their dependence on the characteristics of a "typical" minicomputer can be summarized by Figure 1. The diagram suggests that the partition size (which is generally a function of the primary core size) plays an important role in all aspects of a statistical system design.

Figure 2 gives a schematic representation of some typical implementations (using BASIC or FORTRAN as the source language) that further restricts the space available for active data and system parameters. In general, the use of FORTRAN places much greater constraint on the total size (and hence extensibility) of a system while the most favorable language for modularizing a large system (BASIC with CHAIN and COMMON) is likely to have severe portability problems. The constraints that effect each of several major evaluation items will be elaborated below:

A. User Interface

A.1 Date structure and size of active data

The most distinguishing feature between a statistical system on a minicomputer and one on a maxicomputer is probably the total size of the "active" arrays (variables addressable in the primary core). For a system running on a maxicomputer with a 256K partition size, say, the space allocatable to active arrays generally exceeds the space on a mini allocatable to the entire system. Thus, in order to have the capability of analyzing a moderate to large dataset on a mini (where the raw data must be accessed repeatedly, such as required in various residuals analyses) the system must be

Figure 1

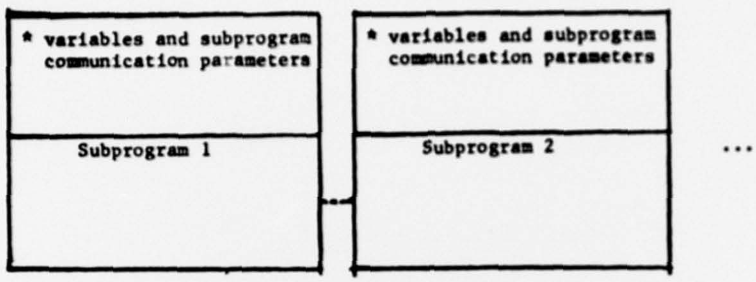
RELATION BETWEEN CONSTRAINTS
AND EVALUATION CRITERIA

Evaluation Criteria	Constraints: WORD LENGTH	Constraints: SOURCE LANGUAGE OPERATING SYSTEM	Constraints: PARTITION SPACE UTILIZATION AND LIMITATION
USER INTERFACE			
Data Structure		X	X
Active Data			X
Command or Control Language			X
Level of Interaction		X	X
Internal Documentation			X
STATISTICAL EFFECTIVENESS			
Versatility			X
Accuracy	X		X
IMPLEMENTATION			
Extensibility		X	X
Portability	X	X	X

Figure 2

EXAMPLES OF SOME TYPICAL IMPLEMENTATION AND PARTITION SPACE UTILIZATION

Standard BASIC (without COMMON and CHAIN capabilities)

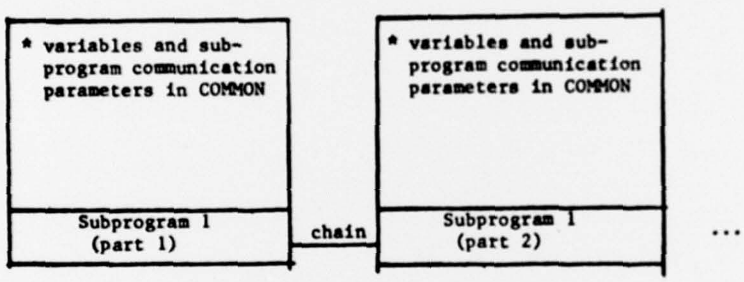


Explicit I/O is required to pass variables and parameters between subprograms

Size of source code for system virtually unlimited

Extensibility: good
Portability: good

BASIC with COMMON and CHAIN (such as HP-2000 BASIC)



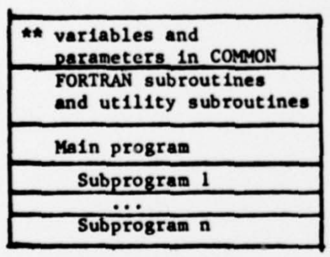
even subprograms can be arbitrarily modularized through COMMON and CHAIN

virtually unlimited size for source programs

Very small portion of partition needed for source

Extensibility: very good
Portability: poor

FORTRAN Load Module (not overlayed)



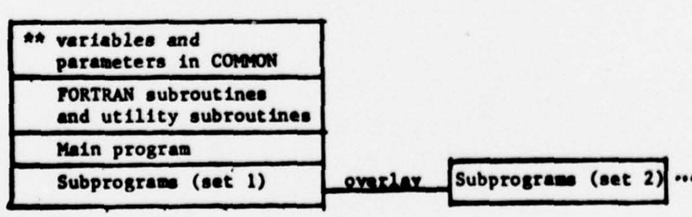
high speed core for data, variables, and system parameters severely limited by size of partition

versatility of system severely limited by the limited amount of space for subroutines

size of source code (function of load module size) limited by size of partition

Extensibility: poor. Lack space. Also, must recompile main program and link
Portability: good if ANSI FORTRAN is used

FORTRAN Load Module (overlayed)



As system grows, more and more FORTRAN subroutines and system utility subroutines must reside in core at all times. This can be accomplished only through a reduction in the size of variables in COMMON. I/O on peripheral device may be necessitated

Extensibility: fair to poor
Portability: almost as good as non-overlayed

* Space relative to partition size remains roughly constant as system grows
** Maximum usable space relative to partition size diminishes as system grows

structured to interface efficiently with data stored in secondary memory locations and devices, whereas a maxi system may have sufficient space to place the entire dataset in core. Moreover, a BASIC system without the COMMON feature will require explicit I/O to pass variables and system parameters among modules or subprograms, thereby exacting a heavy overhead on the performance of the system.

A.2 Command language structure

All interactive systems must have a command language structure. The syntax of the structure may range from simply a dictionary of COMMAND WORDS to one admitting flexible combinations of language phrases and arithmetic expressions. The latter will require a parsing algorithm to interpret the command or control phrases. The partition size of a minicomputer will greatly curtail the space allocatable to the algorithm and thus will limit its complexity and generality.

A.3 Level of interaction between User and System

The minicomputer itself has relatively small effect on this aspect of the software design. The source language used and the mode of communication between the main (driver) program and subroutines (modules) and among modules will determine the efficiency of the interaction (provided the system is optimally designed and coded for man-machine interaction). For example, of the two types of BASIC illustrated in Figure 2, the one with CHAIN and COMMON is much more amenable to a flexible structure for user-system interaction than its counterpart, the standard BASIC .

A.4 Internal documentation

Ideally, the user of an interactive system ought to be able to access all relevant information and documentation about the system on line, without the necessity of a User's Manual or various reference manuals. In practice, no existing system accomplishes this ideal, though some (such as SPEAKEASY, with several hundred pages of text in the HELP file, hierarchically organized in a tree structure) come much closer to an internally-documented system than others. For mini-computers, even considerable less text than that in the SPEAKEASY system would be constrained by the limited partition size. Thus, only the most frequently accessed documentation can be kept in core while the others must be retrieved from secondary or peripheral storage devices.

B. Statistical Effectiveness

B.1 Statistical versatility

The statistical versatility of a system is constrained primarily by the partition space utilization as illustrated in Figure 2, so that the constraint is much more severe for a FORTRAN system than one in BASIC.

A comment is perhaps necessary here to clarify the assertion that a system written in FORTRAN has greater constraints on added statistical capabilities than one written in BASIC. In a FORTRAN environment, statistical as well as I/O tasks that are common to many procedures

(modules) are accomplished by a CALL SUBROUTINE statement within the module with the subroutines being called resident in core at all times. Thus, as a system grows, there will be more and more of such "utility" subroutines. In a BASIC environment, the implicit subroutine call feature does not exist, so that often the identical codes (or codes with different names) are explicitly coded within each and every subprogram or module of the system, as a matter of necessity imposed by the language. In theory, if we simulate this form of inefficiency in FORTRAN (by discarding the effective use of subroutines) then the overlay structure in FORTRAN is no different from the chaining structure in BASIC insofar the programmer is concerned. However, it appears reasonable to assume that when one is working within a portable FORTRAN environment (having sacrificed many non-standard but more powerful features) one is entitled to, and should, make effective use of the SUBROUTINE features in FORTRAN while paying a price in the extensibility of a large system.

B.2 Numerical accuracy

The primary constraint is the word length of a minicomputer which limits the achievable numerical accuracy of the minicomputers. Typically, minicomputers do not have the option to perform computations in double-precision arithmetic while many statistical computational algorithms require double-precision to ensure a high degree of accuracy. A secondary constraint may be considered to be the CPU speed of arithmetic operations because algorithms capable of achieving a high degree of numerical accuracy at the expense of "number crunching" may have to be discarded in favor of less accurate, but much speedier algorithms.

C. Implementation

C.1 Extensibility

The implementation of a system should make allowances for two types of modification or extension:

(1) Added system capabilities (new commands or procedures).

(2) Accommodations of user-supplied procedures or routines.

The feasibility and ease of implementing these depend heavily on the software language used to code the system and to some extent on the operating system on which the package is run. Typically such extensions are much more easily accomplished in BASIC (or any interpretive language) than in FORTRAN (which requires compilation, linking, and the creation of a new load module for the entire system before execution of the new procedure can take place). At the present state of affairs, I would assess the extensibility of a FORTRAN system to be moderately clumsy to fair for the system implementor, and difficult to impossible for the user. On the other hand, extending a system written in BASIC is generally simple and straight-forward.

C.2 Portability

Among all of the evaluation criteria of a statistical system, portability is probably the most challenging one to satisfy as well as one which is much more restrictive than it may seem. The major constraint lies in the fact that even for commonly used languages such as BASIC and FORTRAN, different manufacturers of minicomputers support different features of the languages (as well as compiler and operating system for each of the languages). Consequently, to achieve portability, often certain desirable features

have to be sacrificed (e.g., efficient coding, efficient I/O, and optimal interrupt handling and error recovery) in order that the system can be run without modification on different computers.

5. LOOKING AHEAD TOWARDS THE NEXT GENERATION

In this paper, I presented my impression of the constraints imposed by the present generation of minicomputers on the design and implementation of interactive statistical systems. Given the present rate of technological advances and decrease in the cost of the hardware, it appears likely that the next generation of minicomputers will approach or surpass most of the present generation maxicomputers in capacity and performance. As a result, many of the existing constraints will be partially or totally removed simply as a natural consequence of progress. However, constraints in the portability of software will likely remain in the near future; and may be better or worse in the intermediate future, depending on the demands of the "buyers" and the manufacturers' assessments of the needs of the existing and potential market. In either case, the scientific computing community in general and the statistical computing community in particular (both being small minorities in the computing market of consumers) will be unlikely to have any major impact on the manufacturers' hardware and software designs. Thus, even if it becomes technologically feasible to eliminate all of the constraints discussed in the paper for minicomputers, some of them will remain because of the diversity of demands of different groups of users.

REFERENCES

[1] "All about minicomputers," Feature Report (June, 1972). Datapro Research Corporation, Moorestown, N.J.

[2] "All about minicomputers," Feature Report (September 1975), Addendum 1 (February 1976). Datapro Research Corporation, Dekran, N.J.

[3] Avery, K.R. and Avery, C.A., "Design and development of an interactive statistical system (SIPS)," Proceedings of Computer Science and Statistics: 8th Annual Symposium on the Interface, Health Sciences Computing Facility, UCLA, 1975, 49-55.

[4] Bloomfield, P., "An interactive statistical processor for the Unix timesharing system," Proceedings of Computer Science Statistics: 10th Annual Symposium on the Interface, National Bureau of Standards, Gaithersburg, Maryland, 1977.

[5] Buchness, R. and Engleman, L., "MiniBMD: A minicomputer statistical system," Proceedings of Computer Science and Statistics: 10th Annual Symposium on the Interface, National Bureau of Standards, Gaithersburg, Maryland, 1977.

[6] Fox, D.J., "Some considerations in designing an interactive data analysis system," Proceedings of Computer Science and Statistics: 8th Annual Symposium on the Interface, Health Sciences Computing Facility, UCLA, 1975, 61-5.

[7] Fox, D.J. and Guire, K.E., Documentation for MIDAS, revised 2nd edition, Statistical Research Laboratory, the University of Michigan, August, 1974.

[8] Francis, I., Heiberger, R.M. and Velleman, P., "Criteria in the evaluation of statistical program packages," American Statistician, 29 (February 1975), 52-5.

[9] Guthrie, D., Avery, C., and Avery, K., Statistical Interactive Programming System (SIPS), User's Reference Manual. Oregon State University Bookstore, Corvallis, Oregon, 1974.

[10] Kenny, D.P., Minicomputers. New York: Amacon, 1973.

[11] Ling, R.F., and Roberts, H.V., "IDA and user interface," Proceedings of the Computer Science and Statistics: 8th Annual Symposium on the Interface, Health Sciences Computing Facility, UCLA, 1975, 91-4.

[12] Monrad-Krohn, L., " The micro vs the minicomputer," Mini-Micro Systems (February 1977), 28-33.

[13] Service, J., "Adapting a batch-oriented statistical analysis system to interactive use: the case of SAS," Proceedings of Computer Science and Statistics: 8th Annual Symposium on the Interface, Health Sciences Computing Facility, UCLA, 1975.

[14] SPEAKEASY-3 Reference Manual Level Lambda IBM OS/VS Version, compiled by Cohen, S. and Pieper, S.C., Argonne National Laboratories, Argonne, Illinois, 1976.

[15] Thesis, D.J., "The minicomputer," Datamation (February 1977), 73-82.

[16] Velleman, P. and Welsch, R.E., "Some evaluation criteria for interactive statistical program packages," Proceedings of the Statistical Computing Section, American Statistical Association, 1975, 10-2.

UNCLASSIFIED (14) N84, TR-251

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER N 84	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Constraints in the Design and Implementation of Interactive Statistical Systems for Minicomputers.		5. TYPE OF REPORT & PERIOD COVERED (9) Technical rept., 6. PERFORMING ORG. REPORT NUMBER TR 251
7. AUTHOR(s) (10) Robert F./Ling		8. CONTRACT OR GRANT NUMBER(s) (15) N00014-75-C-0451
9. PERFORMING ORGANIZATION NAME AND ADDRESS Clemson University Dept. of Mathematical Sciences Clemson, South Carolina 29631		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 042-271
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Code 436 Arlington, Va. 22217		12. REPORT DATE (11) May 1977 13. NUMBER OF PAGES 16 (12) 19p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

407 183

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Minicomputer, interactive statistical systems, statistical software design

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this paper, attention is focussed on issues and problems relating to the design and implementation of interactive statistical systems (as opposed to batch systems or small batch or interactive programs) for mini-computers. In particular, constraints imposed by certain characteristics of existing mini-computers (such as size of main storage and data format) as well as related operating systems software and programming languages are discussed. Efforts to relax or eliminate these constraints may be considered as prospects for statistical systems for future generations of mini-computers.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)